Correlation of Cloud Based Computational Fluid Dynamics Simulations to Wind Tunnel Test Results for a NASCAR XFINITY Series Vehicle

Daniel Catranis
Indiana University – Purdue University Indianapolis

ABSTRACT

The cost of setting up and maintaining a high performance computing cluster for large scale CFD usage is too expensive for many smaller motorsport organizations, and so the turn to cloud based computing resources is an attractive one. Cloud based computing centers allow users access to a shared computing cluster and charge based on the amount of resources used by each account. Efficient use of a cloud based computing center necessitates optimizing the CFD simulations to maximize accuracy and minimize cost due to the charge structure in place. This paper attempts to optimize steady state RANS simulations through systematically altering the refinement settings within the simulation mesh. These simulations are conducted using OpenFOAM on two NASCAR XFINITY Series vehicles and are validated using wind tunnel data. The effects of mesh refinement near the surface of the model and the refinement level within a bounding box around the vehicle on the aerodynamic forces of the vehicle are studied and related to the cost of running each simulation. A more computationally intensive transient simulation was also conducted and was not found to have a significant influence on the accuracy of the results beyond that of the steady state simulations.

INTRODUCTION

Given the competitive nature of motorsports, the more prepared a team or organization is the better the chances of them achieving a good result. There are many ways for an organization to prepare for an event, from utilizing simulations to predict vehicle behavior to practicing pit stops or how to repair any number of items that may break down or get damaged. One major type of preparation is to understand the influence of aerodynamics on the vehicle. There are three primary ways to accomplish this. The first is through track testing with a working vehicle. The second is through wind tunnel testing of full size or scale models, and the third is through utilization of Computational Fluid Dynamics (CFD).

Wind tunnel testing and CFD simulations allow an organization to test a large number of setups without having to build full scale components every time something new is tried. While less expensive than prototyping every component, CFD simulations and wind tunnel tests are still costly and regulations limiting the amount of resources spent in these two areas are often regulated by motorsport governing bodies. This is the case in Formula One where the Federation Internationale de l'Automobile (FIA) limits the amount of wind on time in a wind tunnel as well as the total number of Teraflops used in CFD simulations. These limits are often complex and in the case of Formula One, the wind on time and Teraflop usages are linked together where using more of one results in less of the other [1]. In response to the large associated costs and the limited resources, it is important to efficiently utilize CFD simulations.

The use of “fast turn-around CFD simulations” conducted on a single work station was examined by Desai et al where it was found that results accurate to within 25% of those seen in the wind tunnel could be achieved with both 2D and 3D simulations of rear wing components and vehicle underbodies while being completed on a single workstation within a working day [2]. This study was limited to examining a single component of the vehicle without interactions from the rest of the vehicle. Vehicle aerodynamics is highly dependent on every component in the air flow and to fully understand the vehicle performance, a more complex simulation must be conducted.

This paper attempts to expand the “fast turn-around CFD” simulation to a full-scale vehicle model with the aid of cloud-based computing resources. Cloud based computing allows an organization to access high performance computing (HPC) resources without the need to purchase, setup, and maintain a HPC cluster which can cost between five and six figures depending on the cluster specifications [3]. Through the use of cloud-based computing, organizations can gain access to a HPC cluster and are charged based on the resources that they use [4]. This system makes large scale CFD simulations available to smaller teams and organizations without the resources to set up their own HPC cluster. As the utilization cost is directly tied to the amount of resources used, this paper attempts to minimize the computational time required to perform a simulation while also accurately predicting the simulation results.

Simulations are conducted on two NASCAR XFINITY Series vehicles with various levels of mesh refinement to achieve this goal. Simulation results are compared to wind tunnel results gathered by Richard Childress Racing (RCR) to validate the accuracy of the simulation.

SIMULATION SETUP

COMPUTATIONAL RESOURCES – All CFD simulations were conducted using the OpenFOAM simulation software with additional scripts created by TotalSim US to automate some of the OpenFOAM processes. Each simulation was run through the Ohio Supercomputer Center (OSC) [5] and utilized eight nodes and 228 cores on the Owens Cluster. The Owens Cluster is a Dell Intel Xeon E5-2680 v4 machine and each node is comprised of 28 cores and has 128 GB of RAM [6].

An estimation of the cost of each simulation was calculated using Equation 1.

$$Sim\ Cost = RUcost * RU \qquad (1)$$

where $Sim\ Cost$ is the cost of the simulation in dollars, $RUcost$ is the cost associated with each resource unit, and RU is the number of resource units used by the simulation.

VEHICLE MODEL – Two different vehicle models were used in this experiment. The first was a full scale 2016 RCR spec NASCAR XFINITY Series Chevrolet Camaro. The second was a full scale 2017 RCR spec NASCAR XFINITY Series Chevrolet Camaro. Both models were prepared and provided by Chevy Racing for CFD simulation.

There were two major differences between the 2016 and 2017 vehicles. The first was that the front splitter height was lowered by one inch on the 2017 model. This resulted in a new front fascia of the vehicle as well as a lower radiator pan. The second change was a decrease in size of the rear spoiler. Both models were run with a closed tape radiator setup, simulating a fully blocked off radiator. The gap between the front splitter and the ground plane was 19 mm for both models. To achieve this splitter gap on the 2017 model, the front ride height was adjusted to raise the front splitter and account for the one-inch drop between model years. The front tires for each model were positioned with zero degrees of steer. These two models may be seen below in Figures 1 and 2.

Figure 1. 2016 XFINITY Series Vehicle.

Figure 2. 2017 XFINITY Series Vehicle.

FLOW DOMAIN – A rectangular flow domain was used for all simulations. The domain was 107.2 meters long, 27.9 meters wide, and 14.4 meters high. This resulted in a blockage ratio of 0.56%. The location of the vehicle in the domain may be seen below. The distances are marked in terms of the vehicle length (5 meters) and width (1.95 meters). The inlets and outlets are discussed further in the Boundary Conditions section.

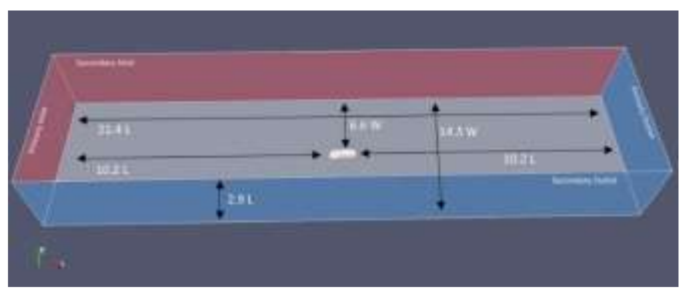

Figure 3. 2016 vehicle model within domain.

BOUNDARY CONDITIONS – The negative X and positive Y faces of the domain were designated as velocity inlets with a constant velocity of 89.408 m/s (200 mph) in the X-direction. A yaw angle of negative four degrees was applied to these inlets and so the actual inlet velocity was 89.19 m/s in the X-direction and 6.24 m/s in the negative Y-direction. The same X and Y velocities were applied along the ground plane to simulate a rolling road condition. The negative Y and positive X faces were designated as pressure-based outlets.

The surface of the vehicle was modelled as a wall and the wheels were modeled using a moving reference frame

(MRF) to simulate the rotation of the tires. Each tire was modelled with a rotation of 259.5 rad/sec to simulate a velocity of 89.4 m/s (200 mph).

TURBULENCE MODEL – As most of the simulations were of the Reynolds Averaged Navier-Stokes (RANS) variety, a turbulence model was necessary to simulate the effects of turbulence in the flow. An SST k-ω model was used which blends two different models depending on the location within the flow. The k-ω model is used near to wall to resolve the boundary layer and the k-ϵ model is used in the far field. A recent study of various turbulence models on a NASCAR vehicle found that this model was able to generate results accurate to within about five percent of those seen in the wind tunnel [7].

MESH – The first step to mesh the domain in OpenFOAM was to create a block mesh of the domain without the vehicle included. The block mesh used for these simulations broke the domain into 96 elements in the X-direction, 32 elements in the Y-direction, and 16 elements in the Z-direction. Once meshed, these initial elements were said to have a refinement level of zero. With the block mesh created, the full mesh was refined with the vehicle included in the domain. OpenFOAM refines meshes in two ways. The first was to specify a distance from the surface of a model and a refinement level for all cells within that distance from the selected surface. The second was to specify a rectangular region of the domain, called a "wakeblock", and a refinement level for all cells within that region. The refinement level indicates how many times the base block mesh cells must be divided in half. A level one refinement divides the level zero elements in half. A level two refinement divides the level one elements in half, and so on.

Initial Mesh – The initial mesh was created using both of the techniques listed above. In addition to the surface level, four additional "distance from surface" refinements were used. These can be seen in Table 1. The initial mesh also made use of three wakeblocks whose parameters may be seen in Table 2.

Table 1. Distance from Surface Mesh Levels

Refinement Region	Distance (m)	Refinement Level
Surface Layer	n/a	9
Region 1	0.04	8
Region 2	0.10	7
Region 3	0.50	6
Region 4	1	5

Table 2. Wakebox Mesh Levels

Wakebox	Min Point (m, m, m)	Max Point (m, m, m)	Refinement Level
Small	-2.7, -1.2, -0.01	4.0, 1.2, 1.8	6
Medium	-3.6, -2.0, -0.01	6.9, 2.0, 2.5	5
Large	-5.5, -3.0, -0.01	30, 3.0, 3.5	4

The initial mesh was made up of 182,479,396 elements for the 2016 model and 182,279,145 elements for the 2017 model. Both models had a minimum y+ value of 9.5 and an average y+ value of 97.5. This was a large y+ value and did not provide adequate resolution to resolve the viscous layer within the boundary layer. A 2008 study by researchers at Cornell University found that in 3D simulations with high y+ values, the downforce could still be predicted reasonably well while the drag values would suffer from increased inaccuracy [2]. This large y+ value was deemed acceptable as the experiment also dealt with the performance trends between different models which would both contain the same inaccuracy and while the absolute values might be off, the trends between models might be consistent with those seen in the wind tunnel. The initial mesh may be seen in Figure 4 with the 2016 vehicle model overlaid.

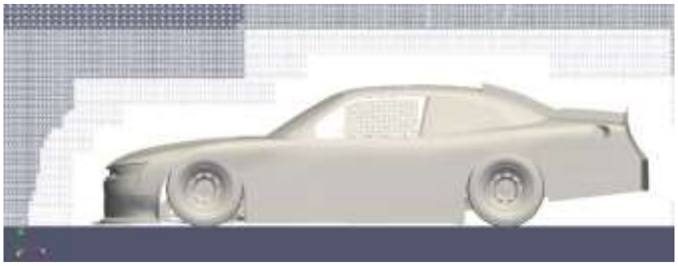

Figure 4. Initial mesh of the 2016 vehicle model.

Mesh Study – Four additional meshes were examined to see how changes to different refinement parameters affected the simulation results. Table 3 shows the changes made to each additional mesh from the initial mesh outlined in Tables 1 and 2.

Table 3. List of Changes from Baseline Mesh to Generate New Mesh

Mesh	Change(s)
2	Region 1 Distance = 0.01m
3	Region 1 Distance = 0.01 m, Small Wakebox Refinement Level = 7
4	Region 1 Distance = 0.02 m, Small Wakebox Refinement Level = 7
5	Region 1 Distance = 0.03 m

The number of elements in each mesh for both the 2016 and 2017 vehicle models are given in Table 4.

Table 4. Element Count for Mesh Variations

Mesh	2016 Elements	2017 Elements
1	182,479,396	182,279,145
2	140,851,754	140,509,402
3	184,445,244	184,042,927
4	195,460,330	195,060,670
5	167,890,052	Not simulated

TRANSIENT ANALYSIS – A transient simulation was also conducted with the 2016 vehicle model and used Mesh 2 detailed above. This simulation was conducted to evaluate the accuracy of a more time intensive simulation against a typical RANS case. The simulation was conducted using a Spalart-Allmaras Delayed Detached Eddy Simulation which applies RANS equations within the boundary layer of the flow and Large Eddy Simulation (LES) equations outside of the boundary layer. The time step used for this simulation was 0.0002 seconds.

WIND TUNNEL COMPARISON

The results from each CFD simulation were compared to wind tunnel test results provided by RCR. These tests were conducted under closed tape conditions, similar to the simulations conducted in this experiment. Further information regarding the wind tunnel tests is confidential to RCR and not discussed here.

To preserve the proprietary nature of the wind tunnel and simulation results, all forces gathered were converted into downforce, sideforce, and drag coefficients through the use of an arbitrary reference area. This was done using Equation 2.

$$C = \frac{2F}{\rho V^2 A} \tag{2}$$

where F is the downforce, sideforce, or drag, ρ is the density, V is the freestream velocity, and A is an arbitrarily chosen reference area.

LIMITATIONS

There are three main limitations in this experiment that were considered. The first was the way in which the mesh was refined. The interior of the models wound up having a high level of refinement due to the fact that the distance from surface refinement method refines the elements in all directions from a surface, not just externally. As the region the driver would occupy has a limited impact on the external flow, it would be acceptable to select a lower refinement level for this region. This is not possible using the standard OpenFOAM tools and ultimately limits the number of elements that are able to be meshed on the outer surfaces.

The second limitation also concerns the mesh and the geometry. The vehicle models were created as thin surfaces and some of the surfaces may have been too thin to accurately get captured by the mesh.

The third limitation was that the CFD simulations were compared solely to wind tunnel results. Wind tunnels are still a simulation of the real world and a full simulation program would validate CFD simulations against wind tunnel results and track testing data.

SIMULATION RESULTS

Results are first presented for the 2016 model year simulations and wind tunnel experiments and then for the 2017 model year simulations. Next, the trends between the 2016 and 2017 models were compared. After analyzing the trends, each 2016 and 2017 simulation was given a single accuracy number by weighting each force based on how important it was to a racing organization. The resources used by each simulation were examined, and finally the tradeoff between accuracy and cost was examined.

WIND TUNNEL CORRELATION – The total downforce, sideforce, and drag coefficients for each of the 2016 simulations are shown in Table 5. Also included are the corresponding coefficients from the wind tunnel. The downforce and sideforce are further broken up into front and rear components in Table 6.

Table 5. 2016 Steady State Simulation Results

Mesh	Downforce	Sideforce	Drag
1	1.061	0.142	0.524
2	0.982	0.137	0.511
3	0.978	0.135	0.508
4	0.984	0.135	0.509
5	1.058	0.140	0.524
Wind Tunnel	0.960	0.249	0.492

Table 6. 2016 Steady State Simulation Front/Rear Results

Mesh	Front Downforce	Rear Downforce	Front Sideforce	Rear Sideforce
1	0.531	0.531	0.124	0.018
2	0.525	0.457	0.126	0.011
3	0.525	0.453	0.124	0.010
4	0.524	0.459	0.124	0.011
5	0.528	0.530	0.124	0.016
Wind Tunnel	0.508	0.452	0.114	0.135

From Tables 5 and 6, it can be seen that downforce and drag were overpredicted when compared to the wind tunnel while sideforce was underpredicted. Looking at the front and rear components of sideforce, it is seen that the front value was overpredicted while the rear value was largely underpredicted. These over/under predictions can be seen in Tables 7 and 8 which show error percentages between the CFD simulations and the wind tunnel experiments.

Table 7. 2016 Steady State Simulation Wind Tunnel Error

Mesh	Downforce	Sideforce	Drag
1	10.6%	-43.1%	6.5%
2	2.3%	-45.0%	4.0%
3	1.9%	-45.9%	3.3%
4	2.5%	-45.7%	3.5%
5	10.2%	-43.6%	6.5%

Table 8. 2016 Steady State Simulation Front/Rear Wind Tunnel Error

Mesh	Front Downforce	Rear Downforce	Front Sideforce	Rear Sideforce
1	4.5%	17.4%	8.5%	-86.9%
2	3.4%	1.0%	9.8%	-91.7%
3	3.4%	0.2%	8.7%	-92.3%
4	3.3%	1.7%	8.3%	-91.7%
5	3.9%	17.3%	8.4%	-87.8%

Each mesh was able to predict the downforce and drag values to within ten percent, with meshes 2, 3, and 4 being within three percent. Each simulation had trouble calculating the sideforce values, and more specifically, the rear sideforce values. While the front sideforce was with ten percent of the wind tunnel values, the rear sideforce for each mesh had an error percentage near 90%. It is believed that this large discrepancy was caused by some of the surfaces in the model being too thin and not being accurately captured within the mesh. One surface of note is the fin running down the left-hand side of the model in Figures 1 and 2.

The total downforce, sideforce, and drag coefficients for each of the 2017 simulations are shown in Table 9. Also included are the corresponding coefficients from the wind tunnel. The downforce and sideforce are further broken up into front and rear components in Table 10.

Table 9. 2017 Steady State Simulation Results

Mesh	Downforce	Sideforce	Drag
1	0.702	0.144	0.433
2	0.640	0.141	0.426
3	0.638	0.139	0.427
4	0.639	0.137	0.425
Wind Tunnel	0.670	0.259	0.411

Table 10. 2017 Steady State Simulation Front/Rear Results

Mesh	Front Downforce	Rear Downforce	Front Sideforce	Rear Sideforce
1	0.364	0.339	0.121	0.024
2	0.356	0.284	0.121	0.021
3	0.357	0.281	0.120	0.019
4	0.355	0.284	0.120	0.017
Wind Tunnel	0.302	0.368	0.114	0.145

From Tables 9 and 10, it can be seen most meshes now underpredict the downforce and sideforce while still overpredict the drag. Looking at the front and rear components of sideforce, it is seen that the front value was still overpredicted while the rear value was largely underpredicted. Unlike the 2016 simulations, the rear downforce was underpredicted. These over/under predictions can be seen in Tables 11 and 12 which show error percentages between the CFD simulations and the wind tunnel experiments.

Table 11. 2017 Steady State Simulation Wind Tunnel Error

Mesh	Downforce	Sideforce	Drag
1	4.8%	-44.4%	5.3%
2	-4.5%	-45.5%	3.7%
3	-4.8%	-46.4%	3.8%
4	-4.7%	-47.2%	3.3%

Table 12. 2017 Steady State Simulation Front/Rear Wind Tunnel Error

Mesh	Front Downforce	Rear Downforce	Front Sideforce	Rear Sideforce
1	20.5%	-8.0%	5.6%	-83.8%
2	17.8%	-22.8%	5.6%	-85.8%
3	18.3%	-23.8%	5.2%	-87.1%
4	17.6%	-23.0%	4.6%	-88.0%

The overall values of downforce, sideforce, and drag show similar discrepancies between CFD simulation and wind tunnel results to those of the 2016 simulations. However, looking at the front and rear downforce values, the front downforce was now overpredicted by twenty percent while the rear downforce was now underpredicted by a similar amount. Once again, the rear sideforce had a large discrepancy between the simulations and wind tunnel results. The reasoning for this discrepancy was the same as given for the 2016 simulations.

PERFORMANCE TRENDS – In addition to analyzing the results from each of the two vehicle models individually, the performance trends between model years for each mesh were examined. The difference between the 2017 and 2016 coefficients for each mesh as well as the wind tunnel results are given in Table 13.

Table 13. Performance Trends between 2016 and 2017 Simulations

Mesh	Front Downforce	Rear Downforce	Front Sideforce	Rear Sideforce	Drag
1	-0.167	-0.192	-0.003	0.006	-0.091
2	-0.169	-0.172	-0.005	0.009	-0.085
3	-0.168	-0.172	-0.004	0.008	-0.081
4	-0.169	-0.176	-0.004	0.006	-0.084
WT	-0.206	-0.084	-0.000	0.011	-0.081

Table 13 shows that the simulations were able to accurately capture the change in performance for the front and rear sideforce as well as the drag between model years. While the individual results for rear sideforce varied greatly from the wind tunnel results, the trend between models was captured as none of the altered components had large impacts on sideforce. The simulations underpredicted the change in front downforce and overpredicted the change on rear downforce between model years.

FORCE WEIGHTING – A weighting was applied to each of the front and rear downforce, front and rear sideforce, and drag error percentages to determine the total error of the solution. The weighting values were selected to place an emphasis on the results that a NASCAR organization would focus on. The selected weightings are given in Table 14.

Table 14. Force Weighting Values

Force	Front Down-force	Rear Down-force	Front Side-force	Rear Side-force	Drag	Total
Weighting	0.2	0.2	0.2	0.2	0.2	1.0

Table 14 shows that each force was assigned the same weighting value of 0.2. This value was selected to place a greater emphasis on downforce and sideforce than on drag. As downforce and sideforce were broken down into front and rear values, the overall downforce and sideforce influence was twice that of drag. The overall error values for the simulations were calculated by multiplying the error percentage for each force by its corresponding weighting and summing the results over each simulation. The overall error is given in Table 15.

Table 15. Overall Simulation Error

Mesh	2016 Error	2017 Error
1	10.02%	12.10%
2	14.69%	16.27%
3	15.35%	16.72%
4	14.98%	17.11%
5	10.31%	N/A

From a pure accuracy standpoint, Table 15 indicates that the initial mesh, Mesh 1, was the mesh that was able to best predict the results seen in the wind tunnel.

COMPUTATIONAL RESOURCES – Cloud computing at OSC charges a user account based on how much computing time was used. This time was measured in actual time, CPU hours, and resource units. Actual time was the amount of time in the real world it took each simulation to run. CPU hours were the total number of hours that all of the processors combined spent running the job. Resource units are the currency that OSC uses to charge user accounts and these units are directly related to the amount of CPU time used. Therefore, it is important to run a simulation using as fine a mesh as is needed to ensure accurate results but no finer in order to minimize computation time. These three values are given for the 2016 simulations in Table 16 and for the 2017 simulations in Table 17.

Table 16. Computing Resources used by 2016 Simulations

Mesh	Actual Time (hours)	CPU Hours (hours)	Resource Units
1	7.67	210.05	172.17
2	5.52	151.09	123.60
3	9.26	253.45	207.47
4	12.17	332.90	272.51
5	8.02	219.49	178.38

Table 17. Computing Resources used by 2017 Simulations

Mesh	Actual Time (hours)	CPU Hours (hours)	Resource Units
1	7.31	199.57	163.81
2	6.00	164.40	134.41
3	9.48	259.40	212.34
4	10.65	291.35	238.49

As shown, the size of the mesh has a direct impact on the amount of time, and subsequently the amount of resource units, necessary to complete a simulation. This trend is shown in Figure 5 which shows the relationship between the number of elements in the mesh and the number of resource units used to complete the simulation.

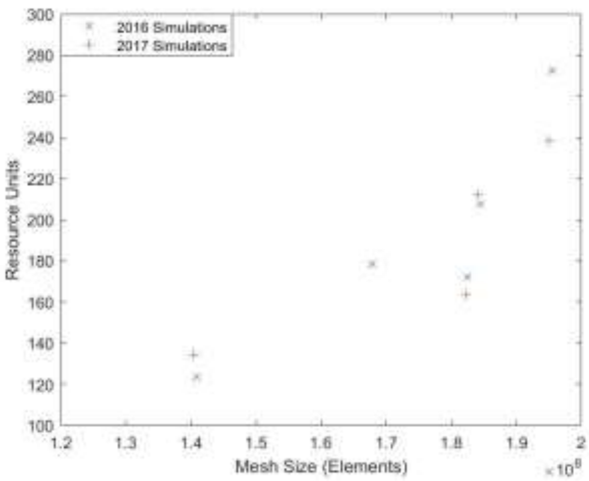

Figure 5. Resource Unit Utilization against Mesh Size

From Figure 5, it can be seen that most simulations followed a linear relationship between mesh size and computational resources utilized. The three outliers were both simulations for the initial mesh, which fell below the trend, and the 2016 simulation for the 4th mesh, which fell above the trend.

From a pure resource utilization standpoint, Tables 16 and 17, as well as Figure 5, indicate that the second mesh required the least amount of resources. This mesh also included the least number of elements.

COST ANALYSIS – Closely associated with the amount of resources used, an estimation of the cost per simulation was calculated using Equation 1. The cost per resource unit is not publicized at OSC and so a value was estimated based off of prior charges and resource unit allocation. The cost factor used for this experiment was \$0.30/resource unit. This factor can be adjusted by the reader to fit a different system. The cost for each simulation is given below in Table 18.

Table 18. Cost per Simulation

Mesh	2016 Simulations (\$)	2017 Simulations (\$)
1	51.65	49.14
2	37.08	40.32
3	62.24	63.70
4	81.75	71.55
5	53.51	N/A

Table 18 shows that the cost per simulation varied greatly depending on the size of the mesh. In the case of the 2016 simulations, two simulations utilizing the second mesh could be run for the same price as one simulation using the fourth mesh. An organization with a limited budget could quickly run out of resources if using a mesh that is too large for the given simulation.

Figure 6 shows the overall error percentage of each simulation plotted against the cost of each simulation. The data points are broken up by mesh.

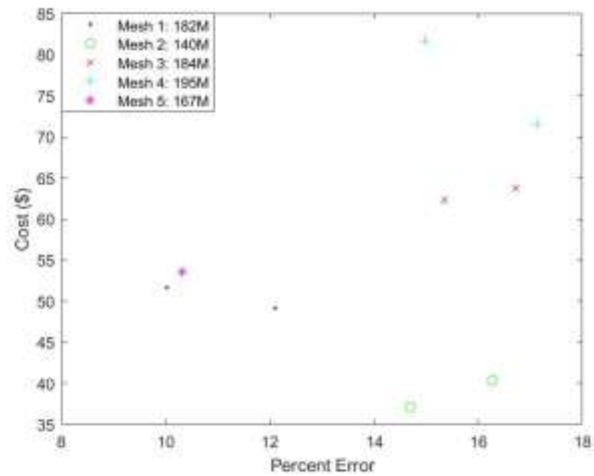

Figure 6. Error Percentage and Cost Comparison

Figure 6 shows that meshes 1 and 5 have low error and medium cost, meshes 3 and 4 have high error and high cost, and mesh 1 has high error and low cost. From this figure, the initial mesh, Mesh 1, would be selected as the optimal mesh settings to use for future simulations. This mesh was selected as it balanced the low error requirement with the low-cost requirement.

TRANSIENT ANALYSIS – The transient simulation was intended to simulate 2.5 seconds of flow over the vehicle. After 0.516 seconds of simulated flow, the simulation was terminated as the OSC user account ran out of disk space to store the simulation results. This highlights a major drawback to cloud based computing, the limited storage available to each user. This storage is often not large enough to conduct large high-level simulations like a transient analysis on a mesh with 140 million elements.

The 0.516 seconds of the completed transient simulation were analyzed and the forces were averaged over the completed time period. The error percentage for the downforce, sideforce, and drag forces are given below in Table 19 and are broken down further into front and rear downforce and sideforce in Table 20.

Table 19. 2016 Transient Simulation Error Percentage

Force	Downforce	Sideforce	Drag
Error	3.77%	-38.03%	-0.01%

Table 20. 2016 Transient Simulation Front/Rear Error Percentage

Force	Front Downforce	Rear Downforce	Front Sideforce	Rear Sideforce
Error	-9.63%	21.62%	19.44%	-91.06%

The transient simulation was able to predict the downforce to a similar accuracy as the steady state simulations and was able to almost identically predict the drag seen in the wind tunnel. As with the steady state simulations, the sideforce was greatly underpredicted, especially in the rear of the vehicle. Using the same weighting values as the steady state simulations, the overall error percentage was 11.93% which was in the same range as the steady state simulations. The actual time, CPU hours, and resource unit utilization are shown below in Table 21.

Table 21. Computing Resources used by 2016 Transient Simulation

Mesh	Actual Time (hours)	CPU Hours (hours)	Resource Units
2	20.38	556.55	456.44

Using Equation 1 and the same cost factor as for the steady state simulations, the cost of this transient simulation was $136.93. The cost was double the cost of the next cheapest steady state run and was only able to run for a fifth of the planned simulation. A NASCAR organization focuses the least on drag, which was the only force to see an increase in accuracy over the steady state simulations. For this reason, the transient simulation was not determined to warrant the increase in computing cost and time utilization.

CONCLUSIONS AND FUTURE CONSIDERATIONS

This paper attempted to determine the feasibility of low cost cloud-based computing for performing CFD simulations on full scale vehicles. Multiple mesh settings were tested on two different NASCAR XFINITY Series Chevrolet Camaros and compared to wind tunnel tests conducted by RCR. The individual results of each simulation were compared to the corresponding wind tunnel results and the deltas between each model year vehicle for the same mesh settings were compared to the deltas seen in the wind tunnel. These results were combined into a single overall error percentage using weighting values assigned to each force and were proportional to how important each force was determined to be relative to a NASCAR program. The cost for each simulation was calculated and compared to the overall error percentage to determine the optimal settings for “fast turn-around CFD.”

To conclude, it was found that the initial mesh, Mesh 1, provided the optimal balance of accuracy and cost. The mesh settings for this mesh had the largest region of fine elements close to the surface of the vehicle and decreased the refinement level within the smallest wakebox. This allowed for more refined elements closer to the vehicle surface which could better capture flow effects from the model.

This study looked at the influence mesh settings had on simulation accuracy and cost, and there are many ways on which to continue to expand this project. The most basic would be to seal off as much of the interior as possible to create a more refined mesh on the outer surface of the vehicle. This would decrease the y+ value and allow for a more accurate resolution of the viscous boundary layer effects.

Continuing with RANS simulations, additional validation could be conducted by comparing the results to not only wind tunnel results, but also to actual track data. This creates a robust system where all three data points are used to validate and improve the vehicle model. Additional simple changes to increase the realism of the model would be to include a steer angle in the tires and to untape the radiator.

As only one transient simulation was conducted in this experiment, an additional study on the effects of transient simulation parameters might prove rewarding. The two areas that could be looked at to achieve accurate transient results are the time step and the simulation length. The time step used in this study was 0.0002 seconds which meant that for every second of simulation time, there were 5000 time steps. If the timestep could be decreased without sacrificing accuracy, the simulation would solve faster, or run further, with no increase in the necessary resources. The length of the simulation might also be an area to study. Finding the minimum time necessary for the solution to accurately converge would also decrease the amount of resources necessary to conduct a simulation. Optimizing both of these parameters would allow for more advanced simulations to be run on a cloud based computing cluster instead of terminating prematurely as the simulation in this study did.

ACKNOWLEDGMENTS

The author would like to acknowledge the support and contributions of Mr. Andrew Borme of the IUPUI Motorsports Department for his insight into motorsports aerodynamics as well as for providing the resources necessary to conduct this project. The author would also like to acknowledge the following individuals. Mr. Noah McKay and Mr. Daniel Honeycutt from Richard Childress Racing for providing the standards with which to set up the simulations to match their wind tunnel experiments. Mr. Kevin Bayless from Chevy Racing for providing the geometry used in these simulations and for the inclusion in multiple wind tunnel tests. Mr. Ray Leto and Mr. Ryan Winfree from TotalSim US for guidance and support on using OpenFOAM as well as the TotalSim specific scripts to run the simulations.

REFERENCES

1. Formula One – Sporting Regulations – 2018, “Restricted CFD (RCFD) Simulations,” Aerodynamic Testing Restrictions 2. Rev. Dec. 2017.
2. Desai, S., Leylek, E., Lo, C., Doddegowda, P., et al., “Experimental and CFD Comparative Case Studies of Aerodynamics of Race Car Wings, Underbodies

with Wheels, and Motorcycle Flows," SAE Technical Paper 2008-01-2997, 2008, doi: 10.4271/2008-01-2997

3. Advanced Clustering Technologies, Inc., "HPS Pricing Guide," http://www.advancedclustering.com/the-cost-of-hpc/, accessed March 2018.
4. Ohio Supercomputer Center, "Charging," https://www.osc.edu/supercomputing/software/general#charging, accessed March 2018.
5. Ohio Supercomputer Center. 1987. Ohio Supercomputer Center. Columbus OH: Ohio Supercomputer Center. http:/osc.edu/ark:/19495/f5s1ph73
6. Ohio Supercomputer Center, "Technical Specifications," https://www.osc.edu/services/cluster_computing, accessed March 2018.
7. Fu, C., Uddin, M., Robinson, C., Guzman, A. et al., "Turbulence Models and Model Closure Coefficients Sensitivity of NASCAR Racecar RANS CFD Aerodynamic Predicitions," SAE Int. J. Passeng. Cars – Mech. Syst. 10(1):2017, doi: 10.4271/2017-01-1547